

Amendments to the Claims

1. (Currently Amended) A computer implemented method for switching fonts without embedding font switches in the data comprising the steps of:

receiving, at the computer, a character to be effectively deleted from a first base font resource;

creating, at the computer, a new font resource that includes the received character; and

linking, at the computer, said new font resource to said first base font resource to in effect delete the received character from said first base font resource, wherein said new font resource and said first base font resource act as if they are a unified font resource.

2. (Currently Amended) The computer implemented method as recited in claim 1 further comprising the steps of:

creating, at the computer, an entry in a first table indicating said new font resource is a second base font resource; and

creating, at the computer, a second and a third table associated with said new font resource, wherein said second table maps code points to glyph indexes, wherein said third table comprises glyphs.

3. (Currently Amended) The computer implemented method as recited in claim 2 further comprising the steps of:

creating, at the computer, a link list in an entry in said first table associated with said first base font resource to link said new font resource to said first base font resource; and

indicating in said entry in said first table associated with said first base font resource to reverse linking of said first base font resource to said new font resource if the received character is a character to be deleted.

4. (Currently Amended) The computer implemented method as recited in claim 3 further comprising the steps of:

receiving, at the computer, an identification of a font resource and a code point; and
transmitting said code point to a rasterizer program associated with said identified font resource.

5. (Currently Amended) The computer implemented method as recited in claim 4, wherein said first base font resource is associated with a fourth table and a fifth table, wherein said fourth table maps code points to glyph indexes, wherein said fifth table comprises glyphs.

6. (Currently Amended) The computer implemented method as recited in claim 5 further comprising the step of:

determining, at the computer, if said code point indexes in said fourth table;
wherein if said code point indexes in said fourth table, then the method further comprises the steps of:
procuring, at the computer, a glyph from said fifth table using a glyph index obtained from said fourth table;
converting, at the computer, said glyph to a bit map representation; and
transmitting said bit map representation from the computer to a printer; and
wherein if said code point does not index in said fourth table, then the method further comprises the step of:
determining, at the computer, if said code point indexes in said second table.

7. (Currently Amended) The computer implemented method as recited in claim 6 further comprising the step of:

determining, at the computer, if said code point indexes in said second table;

wherein if said code point indexes in said second table, then the method further comprises the steps of:

procuring, at the computer, a glyph from said third table using a glyph index obtained from said second table;

converting, at the computer, said glyph to a bit map representation; and

transmitting said bit map representation from the computer to a printer; and

wherein if said code point does not index in said second table, then the method further comprises the step of:

determining, at the computer, if said code point indexes in said fourth table.

8. (Previously Presented) A computer readable medium tangibly embodying programmed instructions which, when executed by a computer system, are operable for performing a method of switching fonts without embedding font switches in the data, the method comprising:

receiving a character to be effectively deleted from a first base font resource;

creating a new font resource that includes the received character; and

linking said new font resource to said first base font resource to in effect delete the received character from said first base font resource, wherein said new font resource and said first base font resource act as if they are a unified font resource.

9. (Previously Presented) The computer readable medium of claim 8 further comprising :

creating an entry in a first table indicating said new font resource is a second base font resource; and

creating a second and a third table associated with said new font resource, wherein said second table maps code points to glyph indexes, wherein said third table comprises glyphs

10. (Previously Presented) The computer readable medium of claim 9 further comprising :

creating a link list in an entry in said first table associated with said first base font resource to link said new font resource to said first base font resource; and

indicating in said entry in said first table associated with said first base font resource to reverse linking of said first base font resource to said new font resource if the received character is a character to be deleted.

11. (Previously Presented) The computer readable medium of claim 10 further comprising :

receiving an identification of a font resource and a code point; and

transmitting said code point to a rasterizer program associated with said identified font resource.

12. (Previously Presented) The computer readable medium of claim 11, wherein said first base font resource is associated with a fourth table and a fifth table, wherein said fourth table maps code points to glyph indexes, wherein said fifth table comprises glyphs.

13. (Previously Presented) The computer readable medium of claim 12 further comprising :

determining if said code point indexes in said fourth table;

wherein if said code point indexes in said fourth table, then the computer program product further comprises the programming steps of:

procuring a glyph from said fifth table using a glyph index obtained from said fourth table;

converting said glyph to a bit map representation; and

transmitting said bit map representation to a printer; and

wherein if said code point does not index in said fourth table, then the computer program product further comprises the programming step of:

determining if said code point indexes in said second table.

14. (Previously Presented) The computer readable medium of claim 13 further comprising :

determining if said code point indexes in said second table;

wherein if said code point indexes in said second table, then the computer program product further comprises the programming steps of:

procuring a glyph from said third table using a glyph index obtained from said second table;

converting said glyph to a bit map representation; and

transmitting said bit map representation to a printer; and

wherein if said code point does not index in said second table, then the computer program product further comprises the programming step of:

determining if said code point indexes in said fourth table.

15. (Previously Presented) A system, comprising:

- a client configured to generate a first data stream comprising page description information;

- a spool coupled to said client, wherein said spool is configured to store said first data stream;

- a resource library configured to store a first base font resource;

- a print server coupled to said spool and said resource library, wherein said print server comprises:

 - a first memory unit operable for storing a printer driver configured to generate a second data stream; and

 - a first processor coupled to said first memory unit; and

 - a printer coupled to said print server, wherein said printer is configured to receive said second data stream generated from said print server, wherein said printer comprises:

 - a second memory unit operable for storing a rasterizer program; and

 - a control unit coupled to said second memory unit;

- wherein said client comprises:

 - a third memory unit operable for storing a computer program for creating a linked resource;

 - a second processor coupled to said second memory unit, wherein said second processor, responsive to said computer program, comprises:

 - circuitry operable for receiving a character to be effectively deleted from a first base font resource;

 - circuitry operable for creating a new font resource that includes the received character; and

 - circuitry operable for linking said new font resource to said first base font resource to in effect delete the received character from said first base font resource, wherein said new font resource and said first base font resource as if they were a unified font resource.

16. (Previously Presented) The system recited in claim 15, wherein said resource library comprises a first table, wherein said second processor further comprises:

circuitry operable for creating an entry in said first table indicating said new font resource is a second base font resource; and

circuitry operable for creating a second and a third table associated with said new font resource to be stored in said resource library, wherein said second table maps code points to glyph indexes, wherein said third table comprises glyphs.

17. (Previously Presented) The system as recited in claim 16, wherein said resource library comprises a first table, wherein said second processor further comprises:

circuitry operable for creating a link list in an entry in said first table associated with said first base font resource to link said new font resource to said first base font resource; and

circuitry operable for indicating in said entry in said first table associated with said first base font resource to reverse linking of said first base font resource to said new font resource if the received character is a character to be deleted.

18. (Original) The system as recited in claim 17, wherein said first processor, responsive to said printer driver, comprises:

circuitry operable for receiving an identification of a font resource and a code point; and

circuitry operable for transmitting said code point to said rasterizer program associated with said identified font resource.

19. (Original) The system as recited in claim 18, wherein said resource library stores a fourth table and a fifth table associated with said first base font resource, wherein said fourth table maps code points to glyph indexes, wherein said fifth table comprises glyphs.

20. (Original) The system as recited in claim 19, wherein said control unit, responsive to said rasterizer program, comprises:

 circuitry operable for determining if said code point indexes in said second table;

 wherein if said code point indexes in said second table, then said control unit, responsive to said rasterizer program, further comprises:

 circuitry operable for procuring a glyph from said third table using a glyph index obtained from said second table;

 circuitry operable for converting said glyph to a bit map representation; and

 circuitry operable for transmitting said bit map representation to a printer; and

 wherein if said code point does not index in said second table, then said control unit, responsive to said rasterizer program, further comprises:

 circuitry operable for determining if said code point indexes in said fourth table.

21. (Original) The system as recited in claim 19, wherein said control unit, responsive to said rasterizer program, comprises:

 circuitry operable for determining if said code point indexes in said fourth table;

 wherein if said code point indexes in said fourth table, then said control unit, responsive to said rasterizer program, further comprises:

 circuitry operable for procuring a glyph from said fifth table using a glyph index obtained from said fourth table;

 circuitry operable for converting said glyph to a bit map representation; and

 circuitry operable for transmitting said bit map representation to a printer; and

 wherein if said code point does not index in said fourth table, then said control unit, responsive to said rasterizer program, further comprises:

 circuitry operable for determining if said code point indexes in said second table.

22-27. (Cancelled)